Machine Learning, 13, 135–143 (1993)

Technical Note
Selecting a Classification Method by Cross-Validation

CULLEN SCHAFFER SCHAFFER@MARNA.HUNTER.CUNY.EDU
Department of Computer Science, CUNY/Hunter College, 695 Park Avenue, New York, NY 10021

Editor: Richard Sutton

Abstract. If we lack relevant problem-specific knowledge, cross-validation methods may be used to select a classification method empirically. We examine this idea here to show in what senses cross-validation does and does not solve the selection problem. As illustrated empirically, cross-validation may lead to higher average performance than application of any single classification strategy, and it also cuts the risk of poor performance. On the other hand, cross-validation is no more or less a form of bias than simpler strategies, and applying it appropriately ultimately depends in the same way on prior knowledge. In fact, cross-validation may be seen as a way of applying partial information about the applicability of alternative classification strategies.

Keywords. Cross-validation, classification, decision trees, neural networks

1. Introduction

Machine learning researchers and statisticians have produced a host of approaches to the problem of classification, including methods for inducing rule sets, linear discriminants, decision trees, Bayesian classifiers, and neural networks. Which of all these should we choose when faced with a practical problem?

Quinlan (1993), reviewing a number of comparative studies, drew two main conclusions. First, no single method or paradigm is uniformly superior. Empirical results amply demonstrate that connectionist methods produce models with greater predictive accuracy for some problems, while statistical and symbolic learning methods prevail in others.

Second, problem-specific knowledge can sometimes help us to guess which method will perform best. As an example, Quinlan draws a distinction between *parallel* learning tasks, in which classification is normally determined by the joint effect of all or most attribute values, and *sequential* tasks, in which few attributes are relevant to the classification of any single case and the relevance of a given attribute depends on the values of others. In general, he suggests, we should prefer connectionist methods for parallel classification problems and symbolic methods for sequential ones.

This is just one example of how prior knowledge can help in selecting an approach to classification. When the underlying relationship is known to be complex relative to the amount of data available for training, Fisher and Schlimmer (1988) and Schaffer (1993) suggest that unpruned decision trees may be more accurate than pruned ones. And if we know that attribute values are heavily affected by noise, two studies cited by Quinlan (Fisher & McKusick, 1989; Shavlik et al., 1991) suggest that neural networks may outperform

decision trees. As Quinlan concludes, our understanding of the applicability of each method is increasing and, as it does, we increase our ability to make use of problem-specific knowledge, when we have it.

Often, however, prior knowledge is unavailable or inconclusive. We may not know, for example, whether the true underlying relationship is more nearly parallel or sequential. In this case, a natural idea is to allow the data itself to indicate which method will work best. We may divide the data into two parts, use one part as input to a number of classification algorithms, and then choose whichever algorithm produces the model most accurate on the second part. Or, taking a more sophisticated approach to the same idea, we may conduct a cross-validation study (Geisser, 1975; Stone, 1974), partitioning the data into a number of groups, using each in turn as a test set for models produced on the basis of the remaining data, and choosing the method that achieves the highest average accuracy.

Here we investigate this latter idea and reach a number of basic conclusions. On the positive side, using cross-validation to select a classification method may yield average predictive performance substantially higher than what could be achieved with any individual method, if the mix of problems includes a reasonable proportion of examples favorable to each. Also, using cross-validation can drastically cut the risk of producing a poor model, since it rarely performs much worse than the *best* of the constituent strategies.

On the other hand, any strategy for inducing models from data amounts to a form of bias, and this is as true for the cross-validation strategy as for the constituent strategies it compares. As a consequence, like any other strategy, the average predictive performance of cross-validation will be better than simple alternatives in some environments and worse in others; and, for a given problem, only domain knowledge can help us decide which approach will be preferable.

In short, cross-validation may lead to better average performance at the same time that it guards against the chance of catastrophic performance, but it does not obviate the knowledge-intensive tack suggested by Quinlan.

2. A cross-validation experiment

This section reports the results of an experiment comparing four classification strategies on five problems. The first three strategies include one for decision trees, one for rule sets, and one for neural networks. The fourth strategy performs a cross-validation study to select one of the first three. The empirical evidence reported here demonstrates the positive aspects of cross-validation noted above. The following section returns to a consideration of some of its limitations.

2.1. Methodology

2.1.1. Classification methods

The experiment of this section compares the performance of three constituent classification strategies with a cross-validation strategy for selecting between them. The three constituent strategies are

- C4.5 (Quinlan, 1986; Quinlan, 1987b), a recent version of the ID3 decision tree induction system, with default parameter settings and pruning;
- C4.5rules (Quinlan, 1987a), a closely related system that produces a rule set from the decision tree induced by C4.5, also with default parameter settings; and
- BP (McClelland & Rumelhart, 1988), a back propagation algorithm for training neural networks. BP was used with one hidden layer of five units and trained for 1000 epochs. The learning rate was set at 1.1 and the momentum at .5.[1]

These three strategies are representative of various classification paradigms, but no attempt was made to represent all major classification paradigms or to choose the best algorithm in each.

A fourth strategy, CV, conducts a 10-fold cross-validation study using training data to compare the three constituent strategies. That is,

- The training data is divided at random into ten equal parts;
- Each of these serves in turn as a test set T. C4.5, C4.5rules, and BP are trained on the remaining data and tested on T;
- The results of the ten tests are averaged and the constituent strategy achieving the highest accuracy is selected; and
- This strategy is run on the full training set to produce a prediction model.

The model produced by CV is thus the one of the three produced by the constituent strategies that cross-validation suggests will be most predictive.

2.1.2. *Test suite*

Five test problems were chosen from the UCI machine learning repository (Murphy & Aha, 1992). Basic information about these is given in table 1; detailed information about how data was transformed or collated in some cases is given in an appendix.

Problems for the test suite were selected on the basis of two criteria, both important in intepreting the results that follow. First, data sets completely unfamiliar to the author were chosen, so that he would have no problem-specific knowledge suggesting which of the four tested strategies was likely to perform best. Second, however, a deliberate attempt

Table 1. The test suite.

		Attributes		
Problem	Size	Discrete	Continuous	Classes
Annealing	898	32	6	5
Glass	214	9	0	7
Image	2320	0	19	7
Sonar	208	0	60	2
Vowels	990	0	10	11

was made to represent in the test suite both problems especially amenable to symbolic methods and problems especially amenable to connectionist methods. On the assumption that data from researchers in symbolic methods dominates the UCI repository, three problems chosen from the main repository directory (Annealing, Glass, and Image) were presumed favorable to symbolic methods. Two remaining problems (Sonar and Vowels), located in a subdirectory and carrying the label "taken from connectionist bench," were presumed likely to favor connectionist methods. Readers familiar with one or more of the test problems might like to guess, on the basis of problem-specific knowledge, which were, in fact, particularly suited to each of the tested classification strategies before proceeding.

2.1.3. Experimental design

Results for the four classification strategies were averaged over ten trials, each conducted using 90% of the data for training and the remainder for testing. Note that this standard cross-validation procedure for measuring the effectiveness of the strategies on fresh data exactly mirrors the preliminary study conduced by CV.

2.2. Hypotheses

Two hypotheses to be tested were formulated as follows before any data were collected:

1. The predictive performance of CV will be nearly as good as the *best* of the other three strategies for each of the five problems.
2. CV's average performance over the test suite will be the best of the four strategies compared.

The second hypothesis is precise as stated; the first lacks a careful specification of what will be considered "nearly as good." In the absence of a natural criterion, it seemed preferable to leave the hypothesis as stated and allow the results to speak for themselves.

2.3. Results

The results of the cross-validation experiment are summarized in table 2. The columns headed Accuracies give the average percentage accuracy achieved by the four classification strategies on each of the test problems and averaged over the suite. The last three columns show how many times in 10 trials CV chose the decision tree, rule set, or neural network model for prediction; in the Glass trials, for example, it chose the tree six times and the rules four.

The last row confirms the prediction that CV would turn in the best average performance. In fact, it outperforms C4.5—the best of the constituent strategies—by an average of 3.5%. CV's superiority to each of the constituent strategies is significant at above the .999 level, using a one-sided paired t test.[2]

Table 2. Cross-validation outperforms constituent strategies.

	Accuracies				CV's Choices		
Problem	Tree	Rules	Net	CV	Tree	Rules	Net
Annealing	92.3 ± 3.3	93.9 ± 2.8	99.0 ± 0.8	99.0 ± 0.8	0	0	10
Glass	66.8 ± 6.1	66.8 ± 5.5	45.7 ± 11.0	65.9 ± 6.1	6	4	0
Image	96.9 ± 1.2	97.1 ± 1.2	89.8 ± 5.2	96.9 ± 1.2	10	0	0
Sonar	69.2 ± 10.2	71.6 ± 11.0	82.6 ± 6.9	80.7 ± 9.5	0	1	9
Vowels	76.5 ± 4.1	72.4 ± 4.1	58.4 ± 5.9	76.5 ± 4.1	10	0	0
Average	80.3	80.0	75.1	83.8			

As noted in the introduction, CV achieves superior average accuracy when it is applied to a mix of problems that includes a reasonable proportion favoring different constituent strategies. Since problems for this experiment were chosen to be unfamiliar to the author, the results demonstrate that it may be profitable to *assume* such a mix and to select a classification method by cross-validation when problem-specific knowledge relevant to the selection is not available.

Of course, the test suite was deliberately constructed to include problems favorable to two different paradigms. In light of this, it is interesting to note that the attempted balancing turned out to be ineffectual and, hence, irrelevant. As table 3 shows, CV turned in the best average predictive performance both for the three problems presumed favorable to symbolic processing and for the two presumed favorable to connectionist processing.

A last main point about the experimental results is that they support the first hypothesis—CV's performance is always within 2% of the best of the other tested strategies. Comparing this relative performance with that of the other strategies, as in table 4, illustrates how effectively cross-validation cuts the risk of highly suboptimal performance. Someone using C4.5 uniformly on the problems of the test suite would stumble badly on the Sonar problem, missing models that accurately classify an additional 13.4% of fresh cases on average. Someone using the tested version of BP would do even worse, missing models

Table 3. Cross-validation performs best even for specialized test suite subsets.

	Average Accuracy			
Problems	Tree	Rules	Net	CV
Glass-Image-Annealing	85.3	85.6	78.2	87.3
Sonar-Vowels	72.8	72.0	70.5	78.6

Table 4. Cross-validation cuts risk.

	Worst Relative Performance	
Method	Gap	Problem
Tree	13.4	Sonar
Rules	11.0	Sonar
Net	21.1	Glass
CV	1.9	Sonar

that accurately classify an additional 21.1% of fresh cases on average for the Glass data. By comparison, CV is extremely safe; in these trials, its average performance is always close to the best observed.

Moreover, for obvious probabilistic reasons, the more at stake, the less likely CV is to choose the wrong model. Looking back at table 2, we see that when there is a large difference in accuracy between the best model and the next best—as in the Annealing, Sonar, and Vowels problems—CV rarely makes the wrong choice. It is only when two models are almost equally good—as for Glass and Image—and distinguishing between them is relatively unimportant that cross-validation begins to falter.

3. Caveats

The most important point to balance against the positive features just illustrated is that, although it may seem natural to distinguish a meta-strategy like cross-validation from direct strategies for classification, this distinction is purely conceptual. In fact, cross-validation simply provides one additional mapping from training sets to models. Any mapping of this kind constitutes an inductive bias; hence, like any other classification strategy, the performance of cross-validation depends on the environment in which it is applied.

The results just presented suggest that there may be practical environments in which cross-validation will outperform other well-known strategies for induction, but the same evidence may be used to illustrate how it might be inferior in other applications. If most problems in an environment were like Sonar, for example, the predictive accuracy of BP would be better than CV's by about two percentage points.

Deciding when cross-validation will yield higher predictive accuracy than a simple alternative amounts to deciding when the implicit bias is appropriate. This is precisely the same kind of decision that Quinlan considered in weighing symbolic and connectionist methods, and it can be made only on the basis of similar kinds of domain-specific information. In the absence of knowledge that the mix of problems we face is more favorable to cross-validation than to a simple alternative, all we can rely on is that it is relatively safe. And, as with all forms of insurance, security carries a cost; if the mix of problems favors a single constituent strategy, the average predictive accuracy of cross-validation must be somewhat worse. See Schaffer (1993) for an extended discussion of this point.

A second caveat is that the degree of security provided by cross-validation depends, among other things, on the number of constitutent strategies. In the experiment reported here, cross-validation was used to select between three classification strategies, but it is trivial to extend the technique to select between 5 or 50. We might like to include neural networks with a range of numbers of hidden units, purely statistical techniques, alternative strategies for decision tree or rule induction, and so on. But, the more poor strategies we inadvertently add, the higher the probability that one of them will appear superior to a much better strategy by pure chance.

A last negative point is that cross-validation necessarily takes many times longer for induction than the *slowest* of the constitutent strategies. For the problems and strategies considered in this note, the cost of cross-validation was not exorbitant—the longest CV runs were under one hour on a Sun 4—but it could easily be prohibitive for some applications.

4. Cross-validation and prior knowledge

The introduction to this note cast cross-validation as an alternative to the use of prior knowledge in selecting a classification method. We have just argued, however, that appropriate application of cross-validation depends on prior knowledge. In fact, the key is the kind of prior information available. If we know we are faced with a parallel classification problem, this information suggests that we should choose a connectionist induction method. On the other hand, if we know only that we will face a stream of problems including both parallel and sequential types, we might prefer a cross-validation strategy.

In general, cross-validation may be viewed as a means of applying *partial* information about appropriate methods for classification. When we know very little about a problem, we may apply cross-validation, as in this note, to select between classification strategies spanning a number of paradigms. When we know more, we may use it to select strategies within a single paradigm—to select the appropriate number of hidden units in a neural network, as is often done, or to select an appropriate degree of pruning in inducing a decision tree, as in the CART program (Breiman et al., 1984).

Cross-validation may also be useful when prior knowledge is suggestive, but not conclusive in selecting a classification method. From a Bayesian point of view, a cross-validation study provides information that should be used to adjust our beliefs. As a practical matter, we may believe that a particular problem is sequential, and hence amenable to decision tree methods, but if the evidence of a cross-validation study suggests that a neural network will be much more predictive, we clearly ought to consider adopting a connectionist approach.

In short, cross-validation and prior knowledge are best seen as complementary. Little has been done to date to help us understand how to apply them together in classification work, and this appears to be an important area for future work.

5. Related work

Some work *has* been done in applying cross-validation in conjunction with prior knowledge in the context of multivariate function estimation (Wahba, 1990). It is not clear whether the results of this work carry over to the problems typically tackled in machine learning research.

Statisticians appear not to have directly addressed the question of the risk protection afforded by cross-validation and its dependence on the number and diversity of constituent strategies. They have, however, undertaken in-depth analyses of other aspects of cross-validation. The theoretical relationship between cross-validation and related methods, including bootstrapping, is discussed by Efron (1982); Stone (1977) gives results on the performance of cross-validation in the long run, as the number of training cases approaches infinity. Wolpert (1992a) attempts, among many other things, an analysis of the conditions under which cross-validation increases predictive accuracy.

Finally, Wolpert (1992b) has advocated using the results of a cross-validation-like study, not to choose a single prediction model, but to combine the predictions of several. Initial results are promising (Breiman, 1992), as might be expected from other work on the value of combining multiple models (Buntine, 1991; Kwok & Carter, 1990; Gams, 1989; Jacobs et al., 1991). With regard to the issues raised here, however, two points are worth keeping

in mind. First, the risk protection afforded by cross-validation can be undermined or even reversed by some combination schemes. Second, although we may conceptually distinguish between simple models and combinations, both are deterministic mappings from attribute vectors to classes. Thus schemes for combining models, however sophisticated, amount to fixed mappings from training sets to the same kinds of predictive models produced by simpler methods and are no less instantiations of bias. Whether combination methods will perform better than simple cross-validation or even the application of a single constitutent strategy depends on the mix of problems to be encountered.

Acknowledgments

Thanks to Sholom Weiss for helping me to understand the merits of cross-validation. Thanks also to Lori Pratt, who provided my copy of BP and extensive tutorial advice on how to use it, and to Ross Quinlan, who provided C4.5.

Notes

1. The author's copy of BP was received with shell scripts that happen to run the program with these parameter settings and they were not adjusted to improve performance on the problems of this experiment. This is not, and is not intended to be, a sophisticated use of neural networks.
2. A paired comparison is essential, since variation between problems would otherwise overwhelm differences between strategies. Standard errors are purposely omitted in the last row, where they would reflect mainly between-problem variation and obscure the highly significant difference in performance between CV and the other strategies.

Appendix: Notes on the data

All data were transformed in three ways for use by BP. Continuous variables were scaled (linearly) to the range [0, 1]. Binary discrete variables—including the class variable for Sonar—were coded using a single variable taking on the values 0 and 1. Multivalued discrete variables—including the class variables for problems other than Sonar—were replaced with a set of binary variables, each one taking on the value 1 for one of the original values and the value 0 for the others.

Annealing. The data set used here is the result of concatenating the files anneal.data and anneal.test in the UCI repository. Missing values indicated by a question mark were treated specially by C4.5 and C4.5rules; for BP these were simply considered an additional discrete value.

Image. The data set used is the result of concatenating the files segmentation.data and segmentation.test in the UCI repository.

Sonar. The data set used is from the file sonar.all-data in the UCI repository.

Vowels. This data consists of measurements of 15 speakers repeating 11 vowel sounds 6 times each. Originally, neural networks were used to classify utterances of the last 7 speakers

on the basis of experience with the first 8. Here the 990 (= 15 × 11 × 6) utterances are divided at random into training and test sets.

References

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Pacific Grove, CA: Wadsworth & Brooks.

Breiman, L. (1992). *Stacked regressions* (Technical Report 367). Berkeley, CA: Department of Statistics, Univerity of California at Berkeley.

Buntine, W. (1991). Classifiers: A theoretical and empirical study. *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence.*

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans*. SIAM.

Fisher, D., & McKusick, K. (1989). An empirical comparison of ID3 and back-propagation. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence.*

Fisher, D., & Schlimmer, J. (1988). Concept simplification and prediction accuracy. *Proceedings of the Fifth International Conference on Machine Learning* (pp. 22–28).

Gams, M. (1989). New measurements highlight the importance of redundant knowledge. *Proceedings of the Fourth European Working Session on Learning* (pp. 71–80). Pitman Publishing.

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association, 70,* 320–328.

Jacobs, R.A., Jordan, M.I., Nowlan, S.J., & Hinton, G.E. (1991). Adaptive mixtures of local experts. *Neural Computation, 3(1).*

Kwok, S.K., & Carter, C. (1990). Multiple decision trees. In R.D. Schacter, T.D. Levitt, L.N. Kanal, & J.F., Lemmer (Eds.), *Uncertainty in artificial intelligence 4.* Amsterdam: North-Holland.

McClelland, J.L., & Rumelhart, D.E. (1988). *Explorations in parallel distributed processing*. Cambridge, MA: MIT Press.

Murphy, P.M., & Aha, D.W. (1992). UCI repository of machine learning databases [a machine-readable data repository]. Maintained at the Department of Information and Computer Science, University of California, Irvine, CA. Data sets are available by anonymous ftp at ics.uci.edu in the directory pub/machine-learning-databases.

Quinlan, J.R. (1986). Induction of decision trees. *Machine learning, 1(1),* 81–106.

Quinlan, J.R. (1987a). Generating productions rules from decision trees. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence.*

Quinlan, J.R. (1987b). Simplifying decision trees. *International Journal of Man-Machine Studies, 27,* 221–234.

Quinlan, J.R. (1993). Comparing connectionist and symbolic learning methods. In S. Hanson, G. Drastal, & R. Rivest, (Eds.), *Computational learning theory and natural learning systems: Constraints and prospects.* Cambridge, MA: MIT Press.

Schaffer, C. (1993). Overfitting avoidance as bias. *Machine Learning, 10,* 153–178.

Shavlik, J.W., Mooney, R.J., & Towell, G.G. (1991). Symbolic and neural learning algorithms: An experimental comparison. *Machine Learning, 6(2),* 111–144.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society (Series B), 36,* 111–147.

Stone, M. (1977). Asymptotics for and against cross-validation. *Biometrika, 64,* 29–35.

Wahba, G. (1990). *Spline models for observational data*. SIAM.

Wolpert, D.H. (1992a). On the connection between in-sample testing and generalization error. *Complex Systems, 6,* 47–94.

Wolpert, D.H. (1992b). Stacked generalization. *Neural Networks, 5,* 241–259.

Received June 4, 1992
Accepted September 23, 1992
Final Manuscript April 6, 1993